

SISTEM MONITORING SKENARIO *TESTING* PADA SISTEM *IFINANCING* STUDY KASUS PT. IMS

Rinda¹, Inge Handriani²

Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Mercu Buana

Jl. Raya Meruya Selatan, Kembangan, Jakarta 11650

E-mail : rinda.ina@gmail.com¹, inge.handriani@mercubuana.ac.id²

Abstrak

Berkembangnya teknologi membuat perusahaan membutuhkan aplikasi untuk membantu bisnis yang dijalankan. Meningkatnya kebutuhan tersebut membuat munculnya konsultan IT yang menawarkan jasa mereka untuk mengembangkan sistem di perusahaan yang membutuhkan sistem. Dalam pengembangan suatu sistem tentu saja harus dilakukan pengujian untuk memastikan sistem yang dibuat sudah sesuai dengan proses yang berjalan saat ini. Pengujian sistem dilakukan untuk mengetahui kualitas sistem yang nantinya akan digunakan oleh penggunanya. Setiap sistem yang diimplementasikan di perusahaan tertentu memiliki perbedaan cara pengujian sehingga dibuatlah skenario *testing* untuk melakukan pengujian tersebut. Skenario *testing* ini tentu saja untuk melihat apakah sistem yang dibuat sudah sesuai dengan kasus - kasus yang mungkin terjadi di lapangan. PT IMS menjalankan skenario *testing* tanpa adanya monitoring sehingga sulit untuk memonitor proses pengujian yang dijalankan. Penelitian ini dilakukan agar proses yang berjalan saat ini dapat termonitor dengan baik sehingga dibuatkan aplikasi skenario *testing*. Aplikasi skenario *testing* dapat memonitor melalui *report* dan grafik pengerjaan yang dilakukan.

Kata kunci : *Monitoring, Skenario testing, Grafik, Forward Chaining*

I. PENDAHULUAN

Didunia bisnis, persaingan antar perusahaan semakin meningkat oleh karena itu setiap perusahaan dituntut untuk melakukan perubahan untuk mencapai kesuksesan melakukan bisnis (Franky Junalfreen T, Yaya Sudarya T, 2018). Persaingan antar bisnis perusahaan atau organisasi membutuhkan dukungan dari industri perangkat lunak yang dapat menyesuaikan kebutuhan pengguna dalam menunjang semua aktifitas bisnis (Wahyu Widodo, 2016). Perusahaan yang mampu bersaing dalam kompetisi bisnis adalah perusahaan yang mampu mengimplementasikan teknologi ke dalam perusahaannya (V.Wiratna Sujarweni, Lila Retnami Utami, 2016). Perusahaan pembiayaan menjadi salah satu pilihan terutama untuk pembiayaan kendaraan (Nur Hari Yansi, Bambang Tetuko, 2015). Berkembangnya perusahaan pembiayaan membuat meningkatnya permintaan aplikasi untuk mengelola bisnis tersebut. Dalam proses perancangan maupun pengaplikasian teknologi informasi, seringkali perusahaan membutuhkan jasa dari konsultan konsultan teknologi informasi/konsultansi TI. Konsultan TI merupakan tenaga yang menyediakan jasa kepenasahatan (*consultancy service*) dalam bidang teknologi informasi (Dewi Paramita, 2015).

PT IMS merupakan konsultan TI yang menghususkan pengembangan aplikasi pembiayaan dengan salah satu nama aplikasinya yaitu *ifinancing*. Setiap aplikasi *ifinancing* yang dikembangkan akan menyesuaikan dengan proses bisnis yang ada di perusahaan pembiayaan tersebut. Ketika implementasi sistem konsultan TI terlalu fokus kepada aspek teknis (pemasangan *software/hardware*) dan finansial dari suatu *project* sistem informasi terkadang tidak memperhatikan faktor - faktor pemakai dari sistem tersebut terutama pada level operasional. Hal tersebut seringkali membuat sistem informasi tersebut kurang sesuai dengan harapan atau waktu implementasi menjadi lebih panjang dari waktu yang sudah ditentukan (Anneke Wangkar, 2014). Oleh karena itu pentingnya kesesuaian antara aplikasi yang dibuat dengan proses operasional yang berjalan.

Pengukuran keberhasilan dan kegagalan dalam menerapkan suatu sistem aplikasi didasarkan pada kepuasan pengguna sistem aplikasi. Agar memperoleh kualitas informasi yang baik, dibutuhkan sistem aplikasi yang baik pula agar data dapat diolah menjadi informasi yang berguna (Agus Widodo, 2016). Selain itu keberhasilan implementasi sistem

informasi dapat terwujud tergantung pada penggunaan sistem yang dijalankan, mudahnya sistem digunakan dan teknologi yang dipakai.

Agar aplikasi yang dikembangkan terjamin kualitasnya maka aplikasi yang sudah dikembangkan sebelum diimplementasikan harus melalui tahap pengujian terlebih dahulu untuk memastikan semua proses berjalan sesuai dengan yang diinginkan. Pengujian dilakukan untuk menemukan kesalahan pada sistem yang telah dikembangkan. Pengujian yang baik adalah dapat menemukan kesalahan yang awalnya tidak ditemukan (M. Sidi Mustaqbal dkk, 2015). Aplikasi yang telah dilakukan pengujian tentunya dapat menjamin kualitas dari aplikasi tersebut.

Pengujian yang dilakukan pada PT. IMS menggunakan metode *black box testing*. *Black box testing* menguji setiap fitur yang diberikan naskah tentang input, proses yang terjadi, hingga bentuk yang diharapkan, dimulai dari kasus positif hingga kasus *negative* (Riad Sahara, Indra Ranggadara, 2018). Pengujian yang dilakukan tentu saja harus termonitor kegiatannya agar mengetahui perkembangan dari sistem yang dilakukan pengujian. Monitoring merupakan sebuah proses penilaian kinerja sistem dari waktu ke waktu (Putu Satya Putra, I Made Sukarsa, I Putu Agung Bayupati, 2017). Saat ini pengujian yang dilakukan di PT. IMS belum dilakukan monitoring sehingga perkembangan proses pengujian tidak diketahui.

Adapun rumusan masalah pada penelitian ini yaitu bagaimana dokumentasi skenario *testing* dan *error* yang ditemukan pada saat menjalankan skenario *testing*, bagaimana proses pemberian tugas skenario *testing* pada orang yang akan mengerjakan *testing*, bagaimana proses pemberian *error* dan pedoman untuk pengerjaan *error* oleh orang yang mengerjakan *error*, bagaimana proses *update progress testing* dan *error* yang sudah dikerjakan, bagaimana memonitor status skenario *testing*.

Batasan masalah pada penelitian ini aplikasi monitoring dirancang untuk mendokumentasikan skenario *testing* dan *error* yang ditemukan, sistem mengakomodir proses pembagian tugas dan mengupdate *progress* pengerjaan, sistem memonitor pekerjaan yang belum dikerjakan dan yang sudah dikerjakan.

Tujuan pada penelitian ini yaitu untuk membangun sistem yang dapat mendokumentasikan skenario *testing* dan *error* yang ditemukan ketika pengerjaan skenario *testing*, membangun sistem yang dapat memberikan skenario *testing* dan *error* skenario yang berisi pedoman pengerjaan, membangun sistem yang dapat mengupdate proses pengerjaan dari skenario *testing* dan *error*, membangun sistem yang dapat memonitor pekerjaan skenario *testing*.

Berdasarkan masalah yang dihadapi saat ini maka dibuatkan sebuah aplikasi sistem monitoring skenario *testing*. Monitoring dilakukan untuk memberi keyakinan bahwa pengendalian yang telah dirumuskan dan terus berjalan sesuai pada jalurnya (Gery Raphael Lusanjaya, 2016). Monitoring skenario *testing* bertujuan untuk memonitoring proses pengerjaan skenario mulai dari input skenario *testing*, *assign* skenario *testing*, proses dokumentasi skenario *testing* dan *error* hingga proses pengerjaan *error* yang dilakukan.

Tabel 1. Penelitian Terdahulu

No	Edisi Jurnal	Perbedaan dengan sistem yang akan dibuat	Persamaan dengan sistem yang akan dibuat
1	ISSN : 2355-6579 Volume : Vol. III No. 1 Tahun : 2016 Judul : Sistem Informasi Monitoring Pengembangan Software Pada Tahap <i>DE</i> velopment Berbasis Web	Sistem monitoring ini menggunakan bahasa pemrograman PHP dan Framework CodeIgniter dengan database MySQL	Membuat sistem monitoring berbasis web yang membahas tentang monitoring suatu permasalahan
2	ISSN : 2477-2062 Volume : Vol. 2 No. 2 Tahun : 2017 Judul : Sistem Monitoring Nilai Siswa Berbasis Android	Sistem monitoring ini berbasis web dan android. Menggunakan bahasa pemrograman PHP untuk web, javascript dan JQuery untuk android serta menggunakan MySQL untuk database	
3	ISSN : 2460-8181 Volume : Vol. 1 No. 1 Tahun : 2015 Judul : Rancang Bangun Sistem Informasi Monitoring Perkembangan Proyek Berbasis Web (Studi Kasus: PT. Inti Pratama Semesta)	Sistem monitoring ini menggunakan teknik perancangan sistem metode OOAD (Object Oriented Analysis Design). Penulis belum membuat bahasa pemrograman serta database apa yang akan digunakan	
4	ISSN : 2528 - 6145 Volume : Vol 1 Tahun : 2017 Judul : Sistem Monitoring Perkuliahan Menggunakan Pola Desain MVC di Politeknik Negeri Madiun	Sistem monitoring ini mengadopsi pola desain MVC (Model-ViewController). Dengan bahasa pemrograman PHP yang menggunakan Framework Phalcon. Database yang digunakan adalah MySQL	
5	ISSN : 2548-964X Volume : Vol. 2 No. 6 Tahun : 2018 Judul : Pengembangan Perangkat Lunak Aplikasi Monitoring Klimatologi Menggunakan Metode RESTful Web service Berbasis Android (Studi Kasus : Stasiun Klimatologi Karangploso Malang)	Sistem monitoring ini menggunakan metode RESTful web service berbasis web dan android. Bahasa pemrograman dibuat dengan PHP dan Framework Code Igniter untuk web dan Java untuk android	

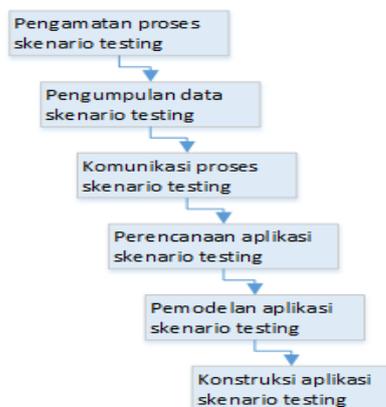
II. PENELITIAN TERDAHULU

Banyak penelitian yang telah dilakukan sebelumnya mengenai aplikasi monitoring berbasis web. Penelitian terdahulu dapat dilihat pada Table 1

Dalam tugas akhir ini penulis akan membuat sebuah sistem monitoring yang dapat mendokumentasikan skenario yang dijalankan dan *error* yang ditemukan saat menjalankan skenario tersebut. Serta adanya monitoring dari pengerjaan yang dilakukan berupa grafik status pengerjaan dan *report* detail dari pengerjaan skenario *testing*. Pada tugas akhir ini sistem dikembangkan berbasis web dengan bahasa pemrograman ASP.NET dan database SQL Server yang berbeda dari penelitian sebelumnya belum ada yang menggunakan bahasa pemrograman tersebut.

III. METODE PENELITIAN

Penelitian ini dilakukan dengan pengamatan langsung di lapangan sebagai salah satu partisipan yang terlibat dalam proses skenario *testing*. Proses pengamatan tersebut kemudian dianalisa dengan menggunakan metode *problem tree* yang merumuskan masalah utama yang kemudian menentukan sebab dan akibat dari proses yang berjalan saat ini. Berdasarkan analisa tersebut maka dibuatlah sistem dengan menggunakan metode *waterfall*. Adapun tahap penelitian dapat dilihat pada Gambar 1



Gambar 1. Tahapan Penelitian

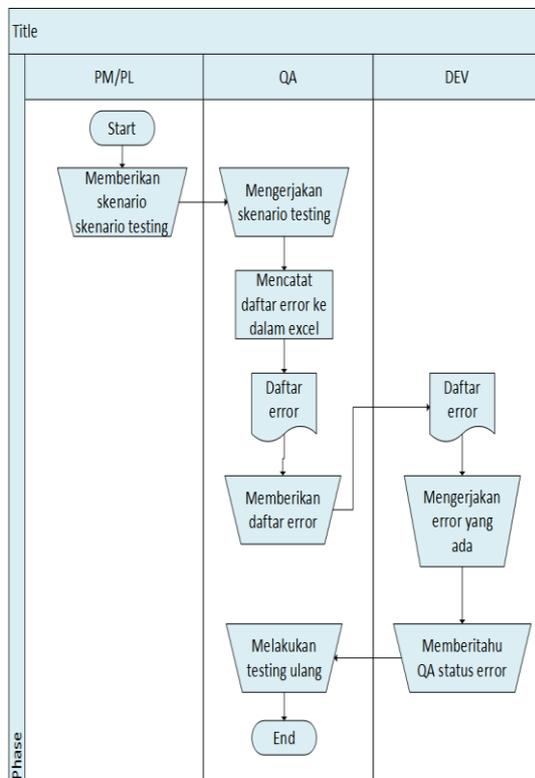
Berdasarkan gambar diatas dapat diuraikan tahapan penelitian aplikasi sistem monitoring skenario *testing* sebagai berikut :

1. Pengamatan proses skenario *testing*
Mengamati secara langsung proses skenario *testing* yang berjalan saat ini dengan cara terlibat langsung sebagai partisipan dari proses skenario *testing* tersebut.
2. Pengumpulan data skenario *testing*
Pengumpulan data skenario yang dijalankan selama di *project* yang sedang berjalan saat ini.
3. Komunikasi proses skenario *testing*
Pada tahap ini dilakukan komunikasi terhadap proses skenario *testing* yang saat ini berjalan di PT.IMS.
4. Perencanaan aplikasi skenario *testing*
Perencanaan terhadap proses skenario *testing* yang akan dibuat menjadi aplikasi monitoring skenario *testing*.
5. Pemodelan aplikasi skenario *testing*
Pada tahap ini akan dilakukan pemodelan terhadap aplikasi skenario *testing* yang akan di buat menggunakan UML(*Unified Modeling Language*) yang terdiri dari *use case*, *Activity diagram* dan *Class diagram*.
6. Kontruksi aplikasi skenario *testing*
Pada tahap ini dilakukan pembuatan aplikasi skenario *testing* menggunakan bahasa pemrograman ASP.Net dan *database* SQL *Server*. setelah aplikasi dibuat maka akan dilakukan proses pengujian dengan *black box testing* untuk memastikan fungsi - fungsi aplikasi dan kesesuaian alur aplikasi dengan proses bisnis yang berjalan.

IV. ANALISA DAN PEMBAHASAN

Komunikasi Data

PT. IMS merupakan konsultan IT yang mengkhususkan diri pada aplikasi *multifinance*. Salah satu product yang ditawarkan yaitu *ifinancing* yang bisa disesuaikan dengan proses bisnis perusahaan yang akan mengimplementasikan sistem tersebut. Proses pengimplementasian tersebut tentu saja harus melewati proses pengujian sebelum digunakan oleh *user*. Prosedur pengujian aplikasi *ifinancing* yang berjalan saat ini dapat dilihat pada Gambar 2



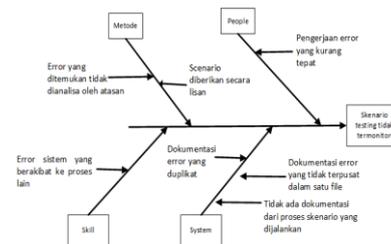
Gambar 2. Proses Bisnis skenario testing

Berdasarkan gambar diatas *Project Manager* (PM) atau *Project Leader* (PL) memberikan tugas kepada *Quality Assurance* (QA) untuk melakukan *testing* aplikasi secara lisan. Kemudian *Quality Assurance* (QA) melakukan *testing* aplikasi yang diberitahukan secara lisan oleh *Project Manager* (PM) atau *Project Leader* (PL). Masing-masing PIC *Quality Assurance* (QA) melakukan dokumentasi *error* yang terpisah melalui *excel*. Daftar *error* yang ada di *excel*

akan dikerjakan oleh *DEVELOPER*. Kemudian *error* yang ada didokumen daftar *error* dikerjakan langsung oleh *DEVELOPER* (DEV). Setelah *DEVELOPER* (DEV) melakukan perbaikan dan memberitahu QA kemudian langsung dilakukan *testing* ulang oleh *Quality Assurance* (QA).

Analisa Sistem Berjalan

Berdasarkan prosedur yang berjalan saat ini maka dapat dilakukan analisa masalah dengan metode *fishbone*. Analisa *Fishbone* dapat dilihat pada Gambar 3



Gambar 3. Analisa Fishbone skenario testing

Berdasarkan *fishbone* maka dapat di uraikan sebagai berikut :

1. Skenario *testing* diberikan secara lisan.
 Solusi : dibuatkan inputan skenario *testing* dalam bentuk web yang kemudian diberikan ke pic yang mengerjakan.
2. Tidak ada dokumentasi dari skenario yang dijalankan.
 Solusi : dibuatkan inputan untuk mendokumentasikan hasil dari skenario *testing* yang dijalankan.
3. Dokumentasi *error* tidak terpusat dalam satu file
 Solusi : dibuatkan inputan dokumentasi *error* dalam bentuk web sehingga dokumentasi terpusat
4. Dokumentasi *error* yang duplikat

Solusi : dibuatkan proses assign sebelum dikerjakan oleh *DEVELOPER*

5. Proses pengerjaan *error* yang kurang tepat.
 Solusi : dibuatkan petunjuk *error* agar *DEVELOPER* lebih tepat dalam mengerjakan *error* sistem
6. *Error* yang ditemukan tidak dianalisa oleh atasan
 Solusi : *error* yang ditemukan harus dianalisa oleh atasan sebelum diberikan kepada *programmer*
7. *Error* yang berakibat ke proses lain
 Solusi : adanya petunjuk yang diberikan oleh atasan kepada *programmer* yang mengerjakan.

Analisa Kebutuhan Sistem

Aplikasi skenario *testing* dibuat berbasis web agar dapat diakses dengan menggunakan internet. Hal tersebut tentu saja untuk memudahkan melakukan monitoring terhadap skenario *testing*.

Aplikasi skenario *testing* yang akan dibuat memiliki fitur sebagai berikut :

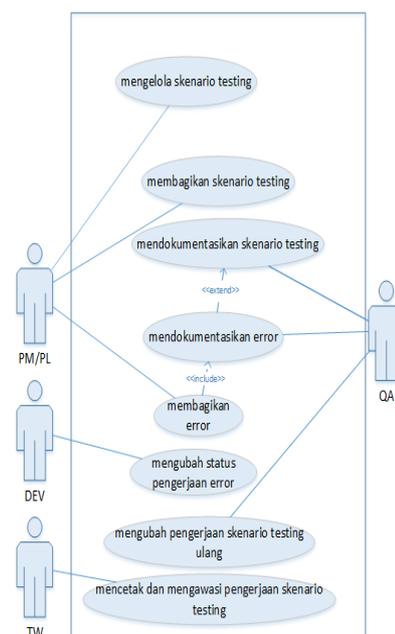
1. *Dashboard* untuk melihat *progress* pengerjaan skenario *testing* hingga pengerjaan *error* yang dilakukan.
2. Daftar *scenario* yang berguna untuk mengelola skenario yang akan diberikan kepada PIC yang ditentukan.
3. *Scenario assign* yang berguna untuk mengelola skenario yang sudah diberikan kepada PIC yang telah ditentukan. Pada skenario *assign* pic akan melakukan dokumentasi skenario dan *error* skenario yang ditemukan.
4. *Error assign* yang berguna untuk mengelola daftar *error* sistem yang ditemukan. *Error* tersebut kemudian dianalisa dan diberikan oleh *PM/PL* kepada pic *DEVELOPER* untuk dikerjakan.
5. *Error list* yang berguna untuk mengelola daftar dari *error* yang sudah ditentukan PIC. Pada fitur ini PIC akan mengupdate *progress error* yang sudah dikerjakan.
6. *Inquiry error* yang berguna untuk melihat daftar *error* dari pengerjaan skenario *testing*.
7. *Print progress* yang berguna untuk mencetak laporan untuk skenario *testing*.

Perancangan aplikasi skenario *testing*

Pada proses pemodelan akan dilakukan perancangan aplikasi skenario *testing* dalam bentuk UML yang nantinya akan menjadi dasar untuk pembuatan aplikasi skenario *testing*.

1. Use Case

Use case ini menggambarkan proses input skenario yang dilakukan oleh *PM/PL* hingga proses pemberian skenario *testing* kepada *QA*. *QA* akan mendokumentasikan hasil *testing* dan juga *error* yang ditemukan. *Error* yang ditemukan tersebut akan dilakukan analisa dan kemudian diberikan oleh *PM/PL* kepada *DEV* yang akan mengerjakan dan mengupdate status pengerjaannya. Setelah *error* tersebut dikerjakan oleh *DEV* maka akan dilakukan retest oleh *QA* dan dilakukan proses update *progress*. Semua kegiatan tersebut akan dicetak oleh *TW* dan dilakukan proses monitoring terhadap skenario *testing* yang dijalankan. Use case skenario *testing* dapat dilihat pada Gambar 4



Gambar 4. Use case aplikasi skenario *testing*

Berdasarkan Gambar 4 maka dapat diuraikan penjelasannya pada Table 2

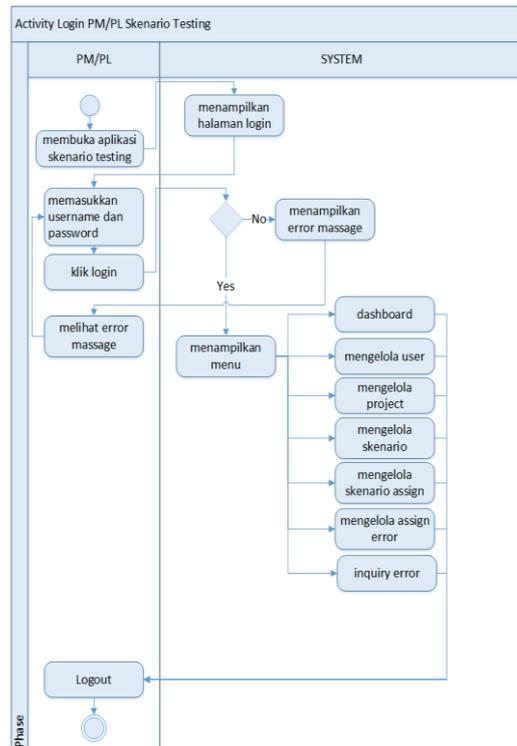
Table 2. Table Use case

Case	Aktor	Deskripsi
mengelola skenario testing	PM/PL	PM/PL menginput langkah - langkah yang harus dilakukan untuk menjalankan skenario testing dan hal - hal yang harus di perhatikan dalam menjalankan proses testing tersebut.
Membagikan skenario testing	PM/PL	PM/PL melakukan assign skenario testing berdasarkan PIC yang telah ditentukan.
Mendokumentasikan Error	QA	QA melakukan dokumentasi dari error yang ditemukan saat menjalankan skenario testing
Mendokumentasikan skenario testing	QA	QA melakukan dokumentasi dari langkah - langkah yang dilakukan pada saat menjalankan skenario testing
Membagikan error	PM/PL	PM/PL melakukan analisa dan assign error yang ditemukan dari proses skenario testing.
Mengubah status pengerjaan error	DEV	DEV melakukan update terhadap error yang sudah dikerjakan
Mengubah pengerjaan skenario testing ulang	QA	QA akan mengupdate progress kembali jika error yang ditemukan sudah diperbaiki oleh DEV
Mencetak dan mengawasi pengerjaan skenario testing	TW	TW melakukan cetak laporan progress harian dan memonitor status pengerjaan error

2. Activity Diagram

a. Activity login PM/PL

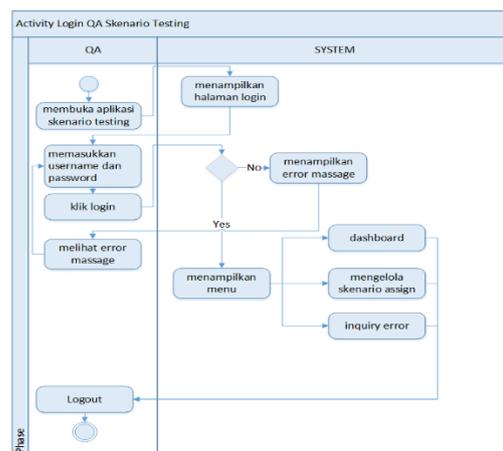
PM/PL membuka aplikasi yang kemudian system akan menampilkan halaman login. Kemudian dilakukan input *username* dan *password* untuk login. Jika login berhasil maka akan masuk ke halaman yang menampilkan menu-menu yang berhak di akses oleh PM/PL. jika tidak berhasil maka akan memunculkan pesan *error* jika salah *password* atau *username* tidak cocok. Activity dapat dilihat pada Gambar 5



Gambar 5. Activity login PM/PL

b. Activity Login QA

QA membuka aplikasi dan kemudian akan tampil halaman login. QA menginput *password* dan *username* kemudian klik login. Jika berhasil maka akan muncul menu yang berhak di akses oleh QA. Activity dapat dilihat pada Gambar 6

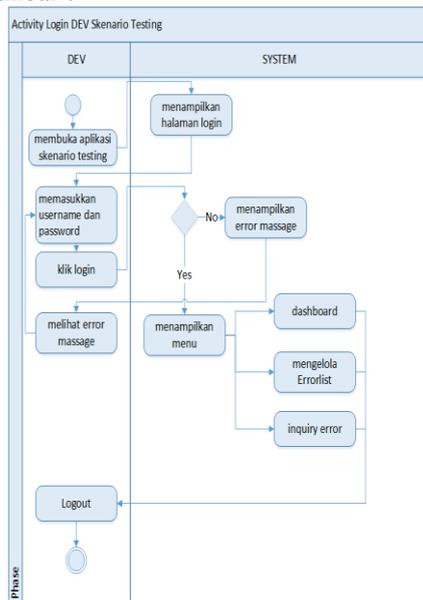


Gambar 6. Activity login QA

c. Activity Login DEV

DEV membuka aplikasi dan kemudian akan tampil halaman login. DEV menginput *password* dan *username* kemudian klik login. Jika berhasil maka akan muncul menu yang berhak di

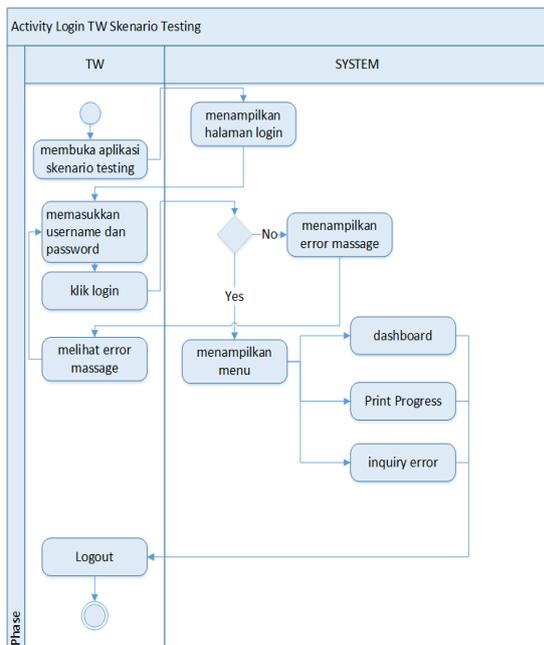
akses oleh *DEV*. *Activity* dapat dilihat pada Gambar 7



Gambar 7. *Activity* login *DEV*

d. Activity Login TW

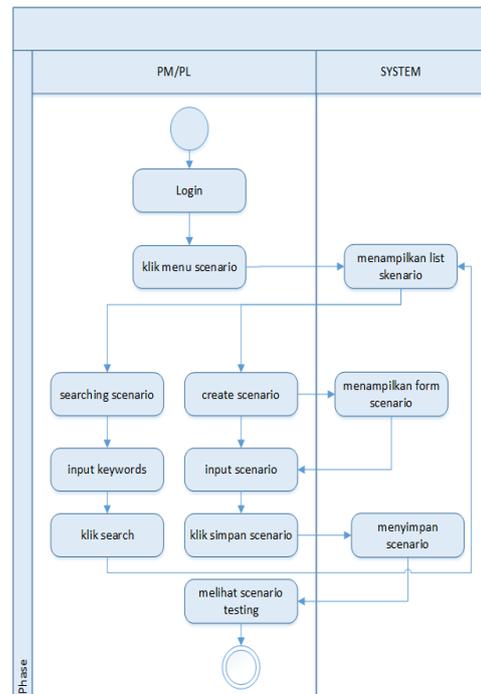
TW membuka aplikasi dan kemudian akan tampil halaman login. *TW* menginput *password* dan *username* kemudian klik login. Jika berhasil maka akan muncul menu yang berhak di akses oleh *TW*. *Activity* dapat dilihat pada Gambar 8



Gambar 8. *Activity* login *TW*

e. Activity mengelola skenario

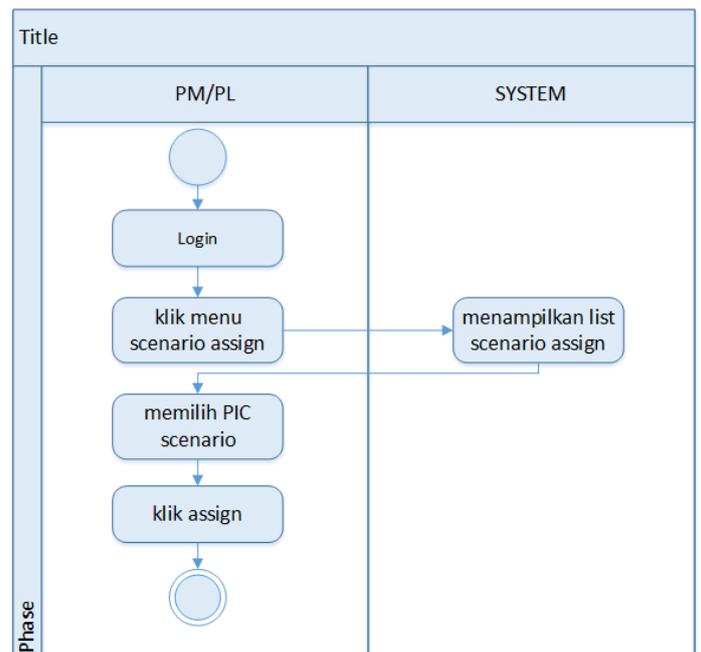
Aktifitas yang dilakukan oleh *PM* pada saat menginput skenario *testing* yang akan dijalankan. *Activity* mengelola skenario dapat dilihat pada Gambar 9



Gambar 9. *Activity* mengelola skenario

f. Activity membagikan skenario testing

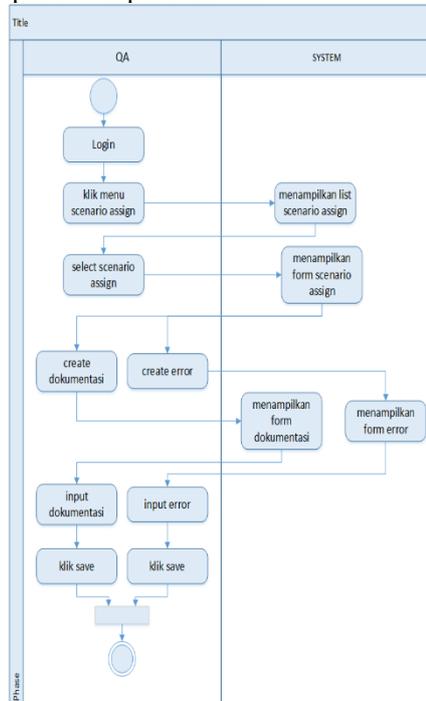
Aktifitas yang dilakukan oleh *PM/PL* pada saat memberikan skenario *testing* kepada *QA*. *Activity* dapat dilihat pada Gambar 10



Gambar 10. *Activity* membagikan skenario *testing*

g. Activity mendokumentasi error dan skenario testing

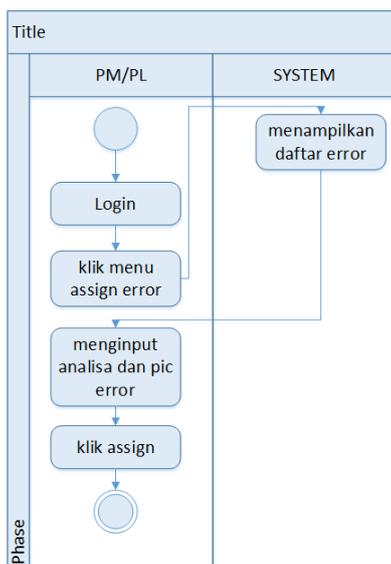
Aktifitas yang dilakukan *QA* setelah menerima skenario *testing* yang diberikan oleh *PM/PL*. *Activity* dapat dilihat pada Gambar 11



Gambar 11. Activity mendokumentasi error dan skenario testing

h. Activity membagikan error

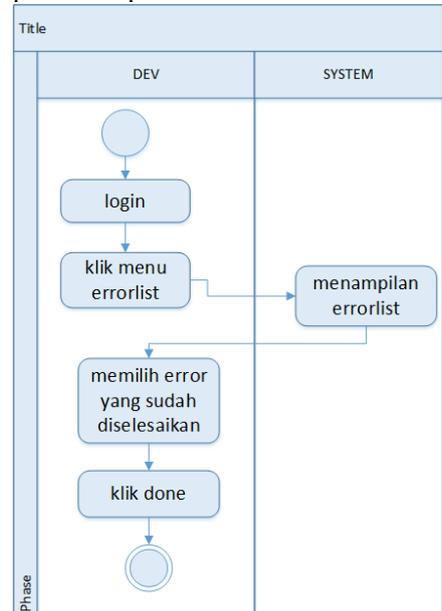
Aktifitas yang dilakukan oleh *PM/PL* saat membagikan *error* kepada *DEVELOPER* yang akan mengerjakannya. *Activity* dapat dilihat pada Gambar 12



Gambar 12. Activity membagikan error

i. Activity mengubah status pengerjaan error

Aktifitas yang dilakukan oleh *DEV* setelah mengerjakan *error* yang diberikan oleh *PM/PL*. *Activity* dapat dilihat pada Gambar 13

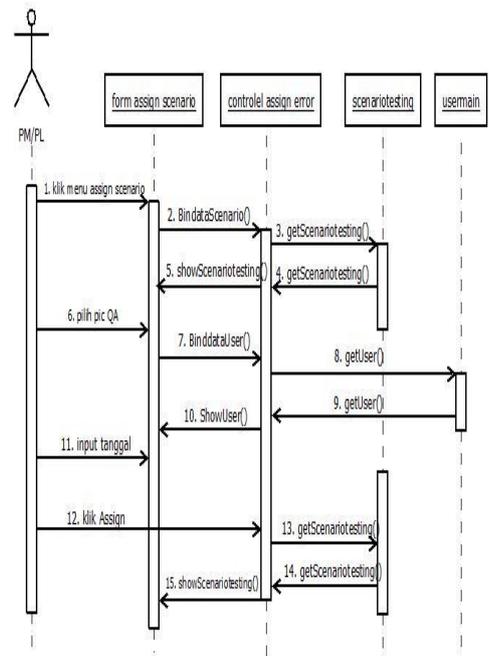


Gambar 13. Activity mengubah status pengerjaan error

3. Sequence Diagram

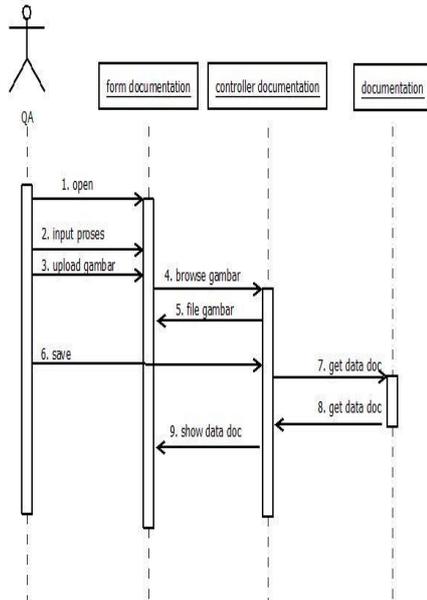
a. Sequence Diagram memberikan skenario testing

Sequence ini menggambarkan proses pemberian skenario *testing* oleh *PM/PL* kepada *QA* yang akan mengerjakan skenario *testing* tersebut. *Sequence* dapat dilihat pada Gambar 14



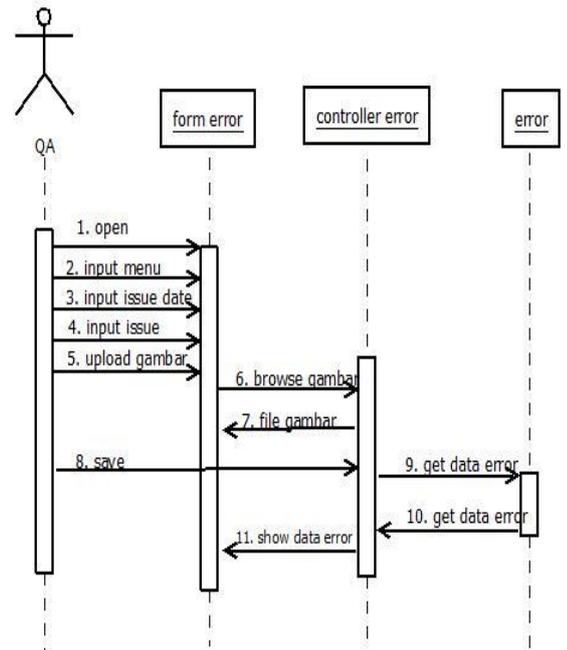
Gambar 14. Sequence Diagram memberikan scenario testing

b. Sequence Diagram mendokumentasikan skenario testing
Sequence diagram ini menggambarkan proses dokumentasi yang dilakukan oleh *QA* ketika menjalankan skenario *testing* yang diberikan. *Sequence* dapat dilihat pada Gambar 15



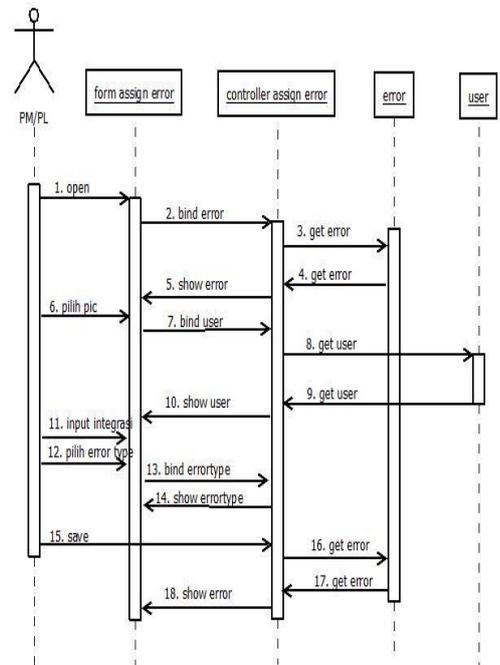
Gambar 15. Sequence Diagram mendokumentasikan skenario testing

c. Sequence Diagram mendokumentasikan error
Sequence diagram ini menggambarkan saat *QA* melakukan dokumentasi *error* yang ditemukan ketika menjalankan skenario *testing*. *Sequence Diagram* dapat dilihat pada Gambar 16



Gambar 16. Sequence Diagram mendokumentasikan error

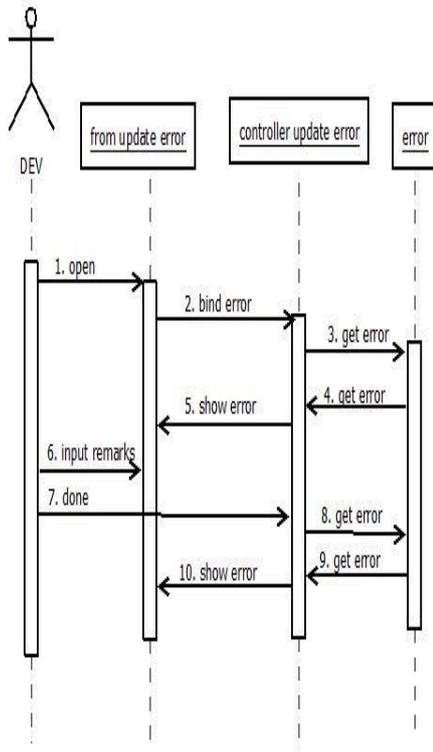
d. Sequence Diagram memberikan error skenario testing
Sequence diagram ini menggambarkan proses pemberian *error* oleh *PM/PL* kepada *DEV* yang akan mengerjakan. *Sequence* dapat dilihat pada Gambar 17



Gambar 17. Sequence Diagram memberikan error skenario testing

e. Sequence Diagram mengubah status pengerjaan error

Sequence diagram ini menggambarkan proses perubahan status yang dilakukan DEveloper terhadap pengerjaan error. Sequence dapat dilihat pada Gambar 18



Gambar 18. Sequence Diagram mengubah status pengerjaan error

4. Class Diagram

Class diagram menggambarkan hubungan antar class yang ada pada aplikasi monitoring skenario testing. Class diagram dapat dilihat pada Gambar 19



Gambar 19. Class diagram skenario testing

Algoritma Forward Chaining

Algoritma forward-chaining adalah satu dari dua metode utama reasoning (pemikiran) ketika menggunakan inference engine (mesin pengambil keputusan) dan bisa secara logis dideskripsikan sebagai aplikasi pengulangan dari modus ponens (satu set aturan inferensi dan argumen yang valid) (Ibnu

Akil, 2017). Ketika mesin inferensi tidak dapat menentukan *object*, ia akan meminta informasi lain. Aturan yang mendefinisikan *object*, membentuk sebuah jalur yang akan mengarah ke *object*. Oleh karena itu hanya satu cara untuk mencapai satu *object* yaitu dengan memenuhi semua aturan (Fajar Masya, Hendra Prastiawan, Sefty Mubaroq, 2016).

Pada sistem monitoring skenario *testing* ini menggunakan algoritma *forward chaining* digunakan untuk menentukan target pengerjaan *error* skenario *testing* berdasarkan *type error* yang ada.

Aturan target berdasarkan *type error* dapat diuraikan sebagai berikut :

1. Jika *Type Error* Validasi Maka maksimal pengerjaan 2 jam
2. Jika *Type Error* Beautification Maka maksimal pengerjaan 1 jam
3. Jika *Type Error* Report Maka 1 hari Maka maksimal pengerjaan 3 jam
4. Jika *Type Error* Proses Maka 1 hari Maka maksimal pengerjaan 1 hari

Beikut contoh penerpan algoritma *forward chaining* berdasarkan aturan diatas.

Declare pengerjaan varchar(10)
 IF (Type error = 'Validasi')

Begin

Set pengerjaan = 2 jam

End

Else if (Type error = 'Beautification')

Begin

Set pengerjaan = 1 jam

End

Else if (Type error = 'Report')

Begin

Set pengerjaan = 1 jam

End

Else if (Type error = 'Proses')

Begin

Set pengerjaan = 1 jam

End

Else

Begin

Raiserror ('waktu pengerjaan tidak ditemukan',16-1)

return

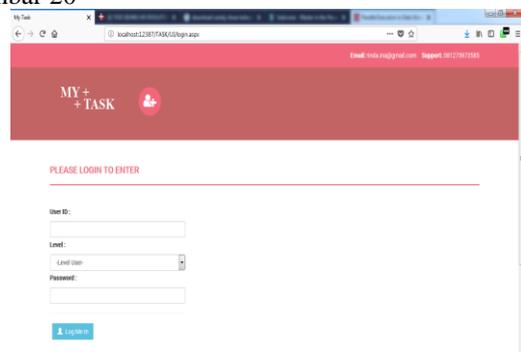
End

Kontruksi aplikasi skenario *testing*

Sistem monitoring skenario *testing* berbasis web yang dibuat menggunakan bahasa pemrograman ASP.NET dan database SQL server. Adapun tampilan dari sistem monitoring skenario *testing* sebagai berikut :

Tampilan Login

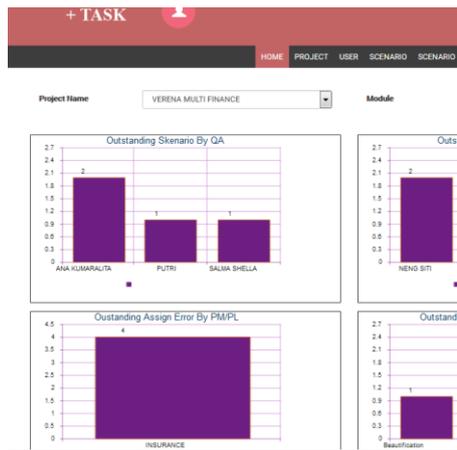
Tampilan login ini berguna masuk ke dalam sistem agar user bisa input,update,delete, ataupun melihat data. Pada form ini user harus menginput user id, level dan *password* yang sesuai agar bisa masuk ke dalam sistem. Tampilan login dapat dilihat pada Gambar 20



Gambar 20. Tampilan halaman login

Tampilan Home

Tampilan home berfungsi untuk menampilkan grafik yang berisi pekerjaan masing-masing individu. Tampilan dapat dilihat pada Gambar 21



Gambar 21. Tampilan home

Tampilan Input Skenario

Tampilan input scenario berfungsi untuk menginput scenario yang akan di *testing*. Tampilan input scenario dapat dilihat pada Gambar 22

The 'SCENARIO INFO' form includes a 'Save' button and a 'Back' button. The 'Project' field is set to 'VERENA MULTI FINANCE' and the 'Module' field is 'FIDUCIA'. The 'Proses Test' field contains a list of 10 steps: 1. Lakukan proses legal handover untuk 5 kontrak; 2. Lakukan proses delete untuk 2 kontrak (tidak jadi diproses handover); 3. Post legal handover; 4. Jalankan proses Legal Request di cabang Legal Custodian yang menerir; 5. Pada Fiducia Process, setelah diapprove akan menimbulkan 1 dokumen; 6. Untuk Fee yang dibayarkan buat melebihi nilai yang dibayar oleh custom; 7. Jalankan pembayaran ke notaris (Sensurunya tidak bisa karena customer); 8. Lakukan penerimaan dari customer di kasir; 9. Lakukan pembayaran ke notaris; 10. Check nilai PPH yang muncul.

Gambar 22. Tampilan input scenario

Tampilan Skenario To Be Assign

Tampilan scenario *to be assign* ini berguna untuk memberikan skenario yang sudah dibuat ke orang yang akan mengerjakan. Pada menu ini akan dipilih siapa yang akan mengerjakan dan target dari penyelesaian skenario tersebut. Tampilan dapat dilihat pada Gambar 23

No	Module Name	Scenario Description	Proses Test	Point Check
1	LEGAL MANAGEMENT	Notary	1. Jalankan proses Legal Request dengan mengahlah notaris 2. Jalankan proses pembayaran DP ke Notaris 3. Lakukan reversal pembayaran DP 4. Post proses Notaris (Seharusnya tidak bisa karena pembayaran DP di reversal) 5. Lakukan pembayaran DP kembali 6. Lakukan posting proses Notaris 7. Pada Settlement input nilai agar Customer menerima kelebihan dan VNF melakukan pembayaran 8. Lakukan proses pembayaran (Check deposit customer) 9. Lakukan reversal (Check deposit customer) 10. Lakukan pembayaran lagi	Cabang yang tidak Le wajib melakukan peng Legal Custodiannya. 7 dokumen Fiducia unt proses legal memerlukan dahulu maka wajib di pembayaran DP ke N dahulu baru bisa dipre kekurangan pembaya dan dibawah dari nilai wajib ditagihkan maka tidak membayar kedu (Ada setting Paramet

Gambar 23. Tampilan skenario *to be assign*

Tampilan Error To Be Assign

Tampilan *error to be assign* ini berfungsi untuk memasukkan siapa yang akan mengerjakan *error* yang ditemukan dari proses *scenario test*, penentuan *type error* yang ditemukan serta jika ada integrasi dari *error* tersebut sebagai petunjuk bagi *developer* ketika mengerjakan *error*. Pada menu ini yang berhak untuk mengakses hanya project manager atau project. Tampilan *error to be assign* dapat dilihat pada Gambar 24

No	Menu	Notes By	Issue	Issue Date	Integrasi	Error Type	PIC DEV	Image
1	Legal Request	ANA KUMARALTA	membuka laptop tidak ada data tapi pada screen sudah terahil	28/05/2018		Beaufication	NENGO SITI	
2	Legal Process	ANA KUMARALTA	data belum batching berdasarkan notary	28/05/2018		Beaufication	NENGO SITI	

Gambar 24. Tampilan *error to be assign*

Tampilan Error On Going

Tampilan *error on going* ini merupakan daftar *error* yang akan dikerjakan oleh *developer*. Pada menu ini *developer* bisa memasukkan catatan dan jika sudah selesai mengerjakan maka akan di update status pengerjaannya menjadi *done*. Tampilan *error on going* dapat dilihat pada Gambar 25

ERROR ON GOING

No	Menu	Notes By	Issue	Issue Date	Integrasi	Error Type	PIC Dev	Target Dev	Remarks	Image
1	Legal Request	ANA KUMARALITA	search berdasarkan process date tidak jalan	28/06/2018		Proses	RINDA	28/06/2018		

Gambar 25. Tampilan error on going

Tampilan Print Progress

Pada menu *print progress* ini dapat mencetak *report* dalam bentuk excel yang terdiri dari tiga jenis *report* yaitu *outstanding* skenario by *QA*, *outstanding error* by *DEV* dan *outstanding assign error* by *PM/PL*. *Report* tersebut ditampilkan berdasarkan nama project, module serta tanggal. Tampilan pada saat ingin mencetak *report* tersebut dapat dilihat pada Gambar 26

PRINT PROGRESS

Gambar 26. Tampilan print progress

V. KESIMPULAN

Proses pengembangan suatu sistem yang disuatu perusahaan sebelum digunakan oleh pengguna tentu saja harus melewati proses pengujian. Pengujian terhadap suatu sistem tentu saja harus dilakukan monitor untuk mengetahui perkembangan dari kualitas sistem yang dihasilkan.

Pada PT IMS monitor tersebut belum terdokumentasi dengan jelas sehingga dibuatkan

aplikasi monitoring skenario *testing* agar dapat memonitor pengerjaan *testing* sistem yang sedang dikembangkan.

Aplikasi skenario *testing* tentu saja menjadi solusi dari proses yang berjalan saat ini sehingga proses pengujian lebih termonitor dengan jelas dan tentu saja jika pengujian dilakukan dengan baik dan termonitor maka akan menghasilkan suatu sistem yang berkualitas yang dapat digunakan oleh pengguna.

DAFTAR PUSTAKA

Agus Widodo, Honorata Ratnawati Dwi Putranti, Nurchayati. (2016). Pengaruh Kualitas Sistem Aplikasi dan Kualitas Informasi terhadap kepuasan pengguna sistem aplikasi RTS (Rail ticketing system) dengan kepercayaan sebagai variabel mediasi (studi pada penumpang “KAI” ekonomi operasi 4 semarang). *Media Ekonomi dan Manajemen*, 31(2), 161.

Andrian Prima Atmaja, Fredy Susanto. (2017). Sistem Monitoring perkuliahan menggunakan pola desain MVC di politeknik negeri madiun,1(1).

Anneke Wangkar. (2014). Efektifitas Implementasi Sistem Informasi Akuntansi Terintegrasi pada PT. ACE HARDWARE INDONESIA TBK. *Jurnal EMBA*, 2(2),1580.

Aprisa, Siti Monalisa. (2015). Rancang Bangun Sistem Informasi Monitoring Perkembangan

- Proyek Berbasis Web Studi Kasus: PT. Inti Pratama Semesta. *Jurnal Rekayasa dan Manajemen Sistem Informasi*,1(1).
- Asti Herlina, Prima Muhammad Rasyid. (2016). Sistem Informasi Monitoring Pengembangan Software Pada Tahap Development Berbasis Web. *Jurnal Informatika*, 3(1).
- Aviator Kevin Budiono. (2017). Evaluasi kemampuan inspeksi pekerja dan penurunan produk cacat di area injection molding PT.X. *Jurnal Titra*, 5(2), 265.
- Darmawan H, Erico dan Laurentius Risal. (2014). Pemrograman Berorientasi Objek C# Yang Susah jadi Mudah. Bandung : Informatika
- Dewi Paramita. (2015). Rancang bangun sistem informasi kolaboratif berbasis web untuk manajemen proyek teknologi informasi. *Jurnal Buana Informatika*,6(3) ,195.
- Erick Kurniawan. (2014). Pemrograman Web Dinamis dengan ASP.NET 4.5. Yogyakarta: Andi Offset.
- Fajar Masya, Hendra Prastiawan, Sefty Mubaroq. (2016). Application Design to Diagnosis of Bone Fracture (Traditioanal) using forward chaining methods. *IRJCS*, 3(9)
- Franky Junalfren T, Yaya Sudarya T. (2018). Analys And Design Of Ordering System Of Building Material And Equipment Case Study : Serba-Jaya Building Shop. *IRJCS*, 5(4).
- Gerry Raphael Lusanjaya. (2016). Akuntabilitas dalam aktivitas monitoring pengendalian intern sistem informasi. *Journal Of Accounting and Business Studies*, 1(1), 89-90.
- M. Sidi Mustaqbal, Roeri Fajri Firdaus, Hendra Rahmadi. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (studi kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3) ,32.
- Nur Hari Yansi, Bambang Tetuko. (2015). Analisis Kinerja keuangan PT Adira Dinamika Multifinance TBK dengan analisis ratio dan analisis du point. *Jurnal Ilmiah Manajemen dan Bisnis*, 1(3) , 1.
- Prakasa Putra Irawan. (2017). sistem pengambilan keputusan untuk menentukan kelayakan penerima kredit mobil di PT. Adira Finance cabang kota pasuruan. *JOINTECS*,1(2),2
- Pressman, R.S. (2015). *Rekayasa Perangkat Lunak: Pendekatan Praktisi* Buku I. Yogyakarta: Andi
- Putu Satya Saputra, I Made Sukarsa, I Putu Agung Bayupati. (2017). Sistem Informasi Monitoring Perkembangan Anak di Sekolah Taman Kanak-kanak Berbasis Cloud. *Lontar Komputer*, 8(2) , 113.
- Riad Sahara, Indra Ranggadara. (2018). Design and Implementation of treasury application based on Mobile in student Organisation Mercubuana University. *IJCSMC*, 7(2).
- Salamun. (2017). Sistem monitoring nilai siswa berbasi android. *RABIT (Jurnal Teknologi dan Sistem Informasi Univrab)*, 2(2).
- Sukamto, Rosa A, dan M.Shalahudin. (2014). *Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek*. Bandung , Informatika.
- Sang Ayu Nyoman Trisna Dewi. (2013). Pengaruh Dukungan Manajemen puncak, kualitas sistem, kualitas informasi, pengguna aktual dan kepuasan pengguna terhadap impelementasi sistem informasi keuangan

- daerah di kota Denpasar. E-Jurnal Akutansi Universitas Udayana, 4(1), 197.
- Wagimin, Helfi Nasution, Novi Safriadi. (2017). Rancang Bangun Sistem Informasi Monitoring Sertifikasi Pegawai Berbasis Web pada PT PLN (Persero) Wilayah Kalimantan Barat. Jurnal Sistem dan Teknologi Informasi, 5(3).
- Wahyu Widodo. (2016). Evaluasi proses pengembangan perangkat lunak pada virtual team Development menggunakan CMMI Versi 1.3. Jurnal Informatika, 10(1) : 1140.
- V Wiratna Sujarweni, Lila Retnami Utami. (2016). Faktor penentu ketepatan waktu corporate internet reporting pada perusahaan manufaktur yang terdaftar di BEI. Jurnal Riset Akuntansi Mercu Buana (JRAMB), 2(1) ,120.